Amendments to the Specification

Please amend paragraphs [0046] and [0053] of the application as follows:

[0046] Embodiments of the present invention require additional steps and/or logic, as compared to known approaches, to scramble the ingress data destined to each sub-group according to a scrambling sequence function, such as a mask. The scrambling sequence function is crafted such that the sum of the bits that toggle in the first sub-group 104 and in the second sub-group 106 is constant, preferably regardless of the ingress data value. In Fig. 8, an embodiment is shown wherein the scrambling sequence ~~functionincludes~~ function includes first and second scrambling patterns 108 and 110, which are applied to the first and second sub-groups, 104 and 106 respectively. The scrambling sequence function is applied to the two sub-groups 104 and 106 so that a constant sum of toggled bits is achieved during every grain, when considering both sub-groups together. In other words, the first sub-group 104, or the left sub-tree, is scrambled according to the first scrambling pattern 108, and the second sub-group 106, or the right sub-tree, is scrambled according to the second scrambling pattern 110. The fact that a constant number of toggled bits is achieved during every grain is solely due to the chosen scrambling patterns. A rule relating to choosing the patterns will be described later.

[0053] Fig. 9 illustrates a flow diagram for describing a device, such as a data transmission apparatus, that incorporates the current management method according to an embodiment of the present invention. The data transmission apparatus can be an output buffered memory switch. Fig. 8 illustrated the egress ports being divided into one group, which was divided into two sub-groups. In contrast, in Fig. 9, the egress ports are partitioned into a plurality of port groups, e.g. P groups, which can advantageously be an even number, though this is not required. As such, embodiments of the present invention can be described as a current demand management scheme that splits the fanout of ingress data from each ingress port to the egress ports into one or more groups. The situation having two groups of egress port groups is shown in Fig. 9. Splitting the fanout into pairs of sub-groups facilitates the scrambling of data going to each sub-group in a complementary fashion. The goal is to ensure that if one sub-group observed M out of 8 bits per byte changing at a particular timeslot, the other sub-group would observe 8-M bits changing. The sum of both sub-groups always adds up to 8 bits regardless of the contents of the data. The sum can be made up of 0

+ 8, or 1 + 7, or 2 + 6, etc. Thus, a certain amount of secondary current balancing can be achieved by having ~~by having~~ more than one egress port group and by applying a different scrambling pattern to each group. By dividing the fanout tree into multiple groups, one can avoid having the entire tree having lop-sided sums, such as 0 + 8, or 1 + 7.